Claims

1. A method for optimizing a system-level simulation of a hardware device, the method comprising the steps of:

providing a system-level model;

dividing the system-level model into a plurality of functional blocks;

providing a mapping between the system-level model and each of the plurality of functional blocks;

compiling each functional block into at least one hardware object; and

linking the at least one hardware object with the system-level model.

2. The method of claim 1 wherein the at least one hardware object is expressed as compiled run-time code.

3. The method of claim 1 wherein each functional block is represented in at least one hardware description language.

4. The method of claim 1 wherein each functional block is represented in at least one high-level language.

5. The method of claim 4 wherein the high-level language comprises at least one of C, C++, SystemC, and Java.

6. The method of claim 1 wherein the at least one hardware object comprises an API.

7. The method of claim 6 wherein the APIs facilitate interface with application-level software programs.

8. The method of claim 6 wherein each object API represents a pin-level interface corresponding to a hardware element.

9. The method of claim 1 wherein the mapping step comprises at least one of the steps of API mapping and abstraction mapping.

10. The method of claim 9 wherein the mapping step comprises API mapping, the API mapping step itself comprising:

receiving input data from the system-level model;

translating the input data into a format readable by at least one hardware object;

providing the input data to the at least one hardware object; and

translating output data from the at least one hardware object into a format readable by the system-level model.

11. The method of claim 10 wherein (i) the system-level model comprises an API presenting an interface accurate with respect to boundaries of a system clock but having system-specific data and access requirements and (ii) the at least one hardware object comprises an API presenting an interface accurate with respect to boundaries of a system clock but having object-specific data and access requirements, the mapping step reconciling the requirements of the system-level API and each object-level API so as to facilitate data interchange therebetween while maintaining adherence to a system clock.

12. The method of claim 9 wherein the mapping step comprises abstraction mapping, the abstraction mapping itself comprising an abstract interface to the system-level model and a pin-level interface to at least one hardware object.

13. The method of claim 12 wherein (i) the system-level model comprises an API presenting an interface accurate with respect to transactions and (ii) the at least one hardware object comprises an API presenting an interface accurate with respect to the boundaries of a system clock, the mapping step reconciling the transaction-based requirements of the system-level API and the clock-

based requirements of each object-level API so as to facilitate data interchange therebetween while maintaining adherence to the system clock.

14. The method of claim 13 further comprising providing a control object, the control object controlling advancement of time and execution of transactions to thereby reconcile the transaction-accurate system-level API with the clock-based object-level APIs.

15. The method of claim 9 further comprising the step of defining a mapping layer.

16. The method of claim 15 wherein the mapping layer comprises a declaration module, a instantiation module, a sensitization module, an initialization module, an execution module, and an output scheduling module.

17. The method of claim 16 wherein the declaration module defines a wrapper module inside the system-level model for accessing the at least one hardware object.

18. The method of claim 16 wherein the instantiation module creates an instance of the at least one hardware object, the at least one hardware object comprising at least one data structure, wherein the at least one data structure receives data from the mapping layer.

19. The method of claim 16 wherein the sensitization module detects a change to a pin of a pin-level interface to the at least one hardware object, the change representing a signal which, if applied to a pin of a hardware element corresponding to the at least one hardware object, would affect an output pin of the hardware element.

20. The method of claim 19 wherein the change comprises assertion of at least one of (i) a clock signal, (ii) an asynchronous reset signal, and (iii) a signal affecting an output pin without requiring toggling of a system clock.

21. The method of claim 16 wherein the at least one hardware object comprises at

least one data structure, wherein the initialization module assigns at least one value to the at least one data structure, thereby initializing the at least one data structure.

22. The method of claim 16 wherein the execution module copies input data from the mapping layer to the at least one hardware object, executes the at least one hardware object in accordance with the input data, and copies output data from the at least one hardware object to the mapping layer.

23. The method of claim 16 wherein the output scheduling module determines when output data is copied from the mapping layer to the system-level model.

24. An apparatus for integrating a system-level simulation and a hardware device, comprising:

a system-level model divided into a plurality of functional blocks, each functional block being represented by at least one hardware object linked to the system-level model; and

a mapping layer between the system-level model and each of the plurality of functional blocks.

25. The apparatus of claim 24 wherein each hardware object is expressed as compiled run-time code.

26. The apparatus of claim24 wherein each functional block is represented in a hardware description language.

27. The apparatus of claim 24 wherein each functional block is represented in at least one high-level language.

28. The apparatus of claim 28 wherein the high-level language comprises at least one of C, C++, SystemC, and Java.

29. The apparatus of claim 24 wherein each hardware object comprises an API.

30. The apparatus of claim 29 wherein the API facilitates interface with application-level software programs.

31. The apparatus of claim 29 wherein the API is a pin-level interface corresponding to a hardware element.

32. The apparatus of claim 29 wherein the mapping layer comprises an API mapping module configured to (i) receive input data from the system-level model, (ii) translate the input data into a format readable by a hardware object, (iii) provide the input data to the hardware object, and (iv) translate output data from the hardware object into a format readable by the system-level model.

33. The apparatus of claim 32 wherein (i) the system-level model comprises an API presenting an interface accurate with respect to boundaries of a system clock but having system-specific data and access requirements and (ii) the hardware object comprises an API that presents an interface accurate with respect to boundaries of a system clock but having object-specific data and access requirements, the mapping layer reconciling the requirements of the system-level API and each object-level API so as to facilitate data interchange therebetween while maintaining adherence to a system clock.

34. The apparatus of claim 33 wherein the mapping layer comprises an abstraction mapping module that itself comprises an abstract interface to the system-level model and a pin-level interface to a hardware object.

35. The apparatus of claim 34 wherein (i) the system-level model comprises an API presenting an interface accurate with respect to transactions and (ii) the hardware object comprises an API that presents an interface accurate with respect to the boundaries of a system clock, the mapping layer reconciling the transaction-based requirements of the system-level API and the clock-based requirements of each object-level API so as to facilitate data interchange therebetween while maintaining adherence to the system clock.

36. The apparatus of claim 35 further comprising a control object, the control object controlling advancement of time and execution of transactions to reconcile the transaction-accurate system-level API with the clock-based object-level APIs.

37. The apparatus of claim 24 wherein the mapping layer comprises a declaration module, an instantiation module, a sensitization module, an initialization module, an execution module, and an output scheduling module.

38. The apparatus of claim 37 wherein the declaration module is configured to define a template of the hardware object to facilitate interface with the system-level model.

39. The apparatus of claim 37 wherein the instantiation module is configured to create an instance of the hardware object, the instance comprising a data structure configured to receive data from the mapping layer.

40. The apparatus of claim 37 wherein the sensitization module comprises means enabling the mapping layer to detect a change to a pin of a pin-level interface of the hardware object, the change representing a signal which, if applied to a pin of a hardware element corresponding to the hardware object, would affect an output pin of the hardware element.

41. The apparatus of claim 40 wherein the change comprises assertion of at least one of (i) a clock signal, (ii) an asynchronous reset signal, and (iii) a signal affecting an output pin without requiring toggling of a system clock.

42. The apparatus of claim 37 wherein the hardware object comprises a data structure for receiving an initialization value from the initialization module.

43. The apparatus of claim 37 wherein the execution module is configured to (i) copy input data from the mapping layer to the hardware object, (ii) execute the hardware object in accordance with the input data, and (iii) copy output data from the hardware object to the mapping layer.

44. The apparatus of claim 37 wherein the output scheduling module is configured to determine when output data is copied from the mapping layer to the system-level model and to thereupon make the output data available to the system-level model.